

GIT & GITHUB

GUÍA DE BOLSILLO

Jesús Arce

info.jess@yahoo.com.ar

Creado: 01 Nov 2019

Actualizado: 17 Mar 2020

Índice

- ¿Qué es GIT?
- ¿Por qué usar un sistema de control de versiones?
- Instalación de GIT
- ¿Qué es GitHub?
- ¿Qué es un repositorio?
- Tipos de repositorios
- Crear un Repositorio Remoto (GitHub)
- Crear un Repositorio Local (PC)
- Agregar nuestra identidad al Repositorio Local
- Conectar Repositorio Local con Repositorio Remoto
- ¿Qué es un commit?
- ¿Qué significa que un archivo está en seguimiento?
- Subiendo archivos a un Repositorio Remoto
- Actualizando archivos de un Repositorio Local
- Clonando archivos de un Repositorio Remoto

➤ ¿Qué es GIT?

Es un **software** de **control de versiones** que **registra** los **cambios** realizados sobre un archivo o conjunto de archivos a lo largo del **tiempo**. De esta forma, podemos recuperar y tener **acceso** a versiones específicas cuando queramos.

➤ ¿Por qué usar un sistema de control de versiones?

Usar un sistema de control de versiones (VSC), te permite **revertir archivos y proyectos enteros** a un estado anterior, **comparar cambios** a lo largo del tiempo, ver **quién modificó** por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más.

➤ Instalación de GIT

- Ir a la [web oficial](#) y descargar el ejecutable
- Ejecutar el archivo que descargamos
- Si tu sistema operativo es Windows, además de instalarse GIT, se instalará en tu máquina una terminal llamada **GIT BASH**
- Una vez instalado GIT, estará disponible el comando `git` para correr en la terminal
- Para verificar que la instalación se haya realizado correctamente, abrir una terminal y correr el comando `git --version`

➤ ¿Qué es GitHub?

GitHub es un **sitio web** en donde podemos **almacenar** los **archivos** y **proyectos** de programación de manera **gratuita**. Para poder hacer uso de sus beneficios sólo hace falta [crearse una cuenta](#) en la plataforma.

➤ ¿Qué es un repositorio?

Es el lugar donde se irán **almacenando** los **archivos** de nuestro proyecto. En GitHub podemos tener la cantidad de proyectos que queramos, en donde a **cada proyecto** le corresponderá **un repositorio**.

➤ Tipos de repositorios

Los repositorios que se alojan en GitHub los llamamos Repositorios **Remotos**, mientras que a los que se alojan en nuestra PC los llamamos Repositorios **Locales**. Es necesario crear un

vínculo entre ambos para poder mantener **actualizados** los archivos locales que están conectados a ese repositorio en la nube.

➤ **Crear un Repositorio Remoto (GitHub)**

- Una vez iniciada la sesión en GitHub, dar click al icono + sobre la barra principal y elegir la opción **New repository**
- Veremos en pantalla un formulario
- Completar únicamente el nombre que le queremos dar a nuestro repositorio. Podemos nombrarlo como queramos, pero debe ser un nombre que no hayamos usado para otro repositorio
- Ir hacia abajo de todo y apretar el botón **Create Repository**

➤ **Crear un Repositorio Local (PC)**

- Crear una carpeta en nuestra PC, que será donde alojaremos nuestro proyecto. Esta carpeta será nuestro **repositorio local**
- Dentro de la carpeta, abrir una terminal y correr el comando `git init`.
- Este comando **inicializa un repositorio local** en la carpeta del proyecto

➤ **Agregar nuestra identidad al Repositorio Local**

Para que git pueda hacer un **completo seguimiento** de los **cambios** realizados, necesitamos decirle al repositorio quien somos.

- Abrir una terminal en la ubicación de nuestro repositorio local
- Correr el comando `git config user.name "nombreDeUsuario"`. en donde, entre comillas, debemos escribir nuestro nombre de usuario tal cual aparezca en GitHub
- Para verificar que ingresamos bien nuestro nombre de usuario, correr el comando `git config user.name` y presionar *enter*
- Correr el comando `git config user.email "nombre@email.com"`. en donde, entre comillas, debemos escribir el email con el que nos registramos en GitHub
- Para verificar que ingresamos bien nuestro email, correr el comando `git config user.email` y presionar *enter*

Para configurar nuestra identidad de manera global y no tener que estar aclarando siempre nuestro email y nombre de usuario, agregar la palabra `--global`.

- `git config --global user.name "nombreDeUsuario"`.
- `git config --global user.email "nombre@email.com"`.

➤ Conectar Repositorio Local con Repositorio Remoto

Para que nuestro Repositorio Local sepa a donde queremos subir nuestros archivos hace falta especificarlo.

- Tener creado previamente un repositorio en GitHub
- Ir a la ubicación del mismo y copiar la url
- Escribir el comando `git remote add origin`.
- Pegar la **url** después de la palabra **origin** (*dejando un espacio de por medio*) y presionar enter
- Para verificar que el paso anterior se ejecutó correctamente, correr el comando `git remote -v`. Deberías ver en la terminal la palabra `origin`, seguida de la url

➤ ¿Qué es un commit?

Cada vez que subimos archivos (*nuevos o modificados*) a un Repositorio Remoto, se suben en forma de un pequeño **paquete de archivos**. Cada paquete tiene una **fecha de creación** (*timestamp*) y un **autor**.

Es a través de los **commits** que vamos a poder hacer el seguimiento de los cambios que se van realizando en los proyectos, ya que cada uno de ellos genera un **punto cronológico** en la línea del tiempo del proyecto.

➤ ¿Qué significa que un archivo está en seguimiento?

Cuando enviamos un archivo al repositorio, estamos diciéndole a GIT que queremos hacerle un **seguimiento** al mismo a través del tiempo.

Es decir, queremos que se guarde el **estado actual** de ese archivo, para que cada vez que hagamos un cambio nuevo y lo enviemos, podamos **comparar estados** y ver cómo estaba en determinado momento. **Seguirlo** a lo largo del proyecto.

➤ Subiendo archivos a un Repositorio Remoto

Para poder subir nuestros archivos a la nube, debemos seguir los siguientes pasos:

- Abrir una terminal en la ubicación de nuestro repositorio local
- Correr el comando `git status` para ver el **estado** de nuestros archivos (*aquellos en rojo son los archivos que aún no están en seguimiento*)
- Correr el comando `git add` para indicar que queremos **agregar todos** los archivos al repositorio

- Para **agregar un sólo archivo**, correr el comando `git add archivo.extension` en donde deberemos indicar tanto el nombre como la extensión del archivo
- Correr el comando `git status` para ver el **estado** de nuestros archivos (*aquellos en verde son los archivos que serán agregados al repositorio, por lo tanto estarán en seguimiento*)
- Para **confirmar** que queremos subir de manera definitiva aquellos archivos que agregamos, correr el comando `git commit -m "mensaje"` en donde, entre comillas, deberemos escribir, en lo posible, un mensaje corto que resuma el trabajo que estamos subiendo
- Para **enviar** los archivos al repositorio remoto correr el comando `git push origin master`

➤ Actualizando archivos de un Repositorio Local

Para poder actualizar los archivos de nuestro repositorio local con respecto a los que están en el repositorio remoto, debemos correr el comando `git pull origin master`.

➤ Clonando archivos de un Repositorio Remoto

Para **descargar** por primera vez un repositorio remoto a nuestra máquina, tendremos que **clonarlo**.

- Abrir una terminal en la ubicación en donde queramos clonar el proyecto
- Copiar la **url** del repositorio que queremos clonar
- Escribir el comando `git clone`.
- Pegar la **url** después de la palabra clone (*dejando un espacio de por medio*) y presionar enter